Manual && Reproduction Report for

Watch and Act: Multi-orientation Open-set Scene Text Recognition via Dynamic Expert Routing

Hardware requirements

GPU:

Testing: NVIDIA GPU with Turing or later structure and more than 4 Gib of vram.

(yes pascals, maxwells or even fermis may work, but you gonna need to know what you are doing)

Training: 24GiB GPU with Turing or later structure for regular model (dunno if the large one fits)

CPU: X86 cpus with AVX2 instruction set

(CPUs without AVX2 may or may not work with different version of torch, I don’t know)

RAM: 16GiB (Testing) 32Gib (Training)

Disk: 200GiB

Internet: Yes

Software Requirements

1. Fresh installed Archlinux with these packages:
sudo vim plasma-meta nvidia openssh firefox dolphin konsole wget git less

2. The user name is set to lasercat and it has to be a sudoer (or you may need to go thorugh the code to replace paths if something went south).

3. ***Keep important data off this device!***

I don’t want to wipe your data due to one or two failed cd commands followed by mv and/or rm

Trivia: Manul, aka the Pallas’ cat, is the oldest cat specie still alive on the earth.

Environment setup

1. install the following packages once you boot in.

sudo pacman -Syu pycharm-community-edition plasma-meta python-paramiko python-lmdb python-numba python-opencv python-pillow python-pip python-pyqtgraph python-pytorch-opt-cuda python-regex python-scikit-learn python-scipy python-torchvision-cuda python-tqdm python-xmltodict make gcc cmake unzip python-setuptools

2. make dirs

mkdir ~/cat ~/ssddata ~/hydra_saves ~/ssddata/anchors

3. setup python stuff

mkdir ~/catvenv/; python3 -m venv ${PWD}/catvenv --system-site-packages;

source ~/catvenv/bin/activate; pip install easydict editdistance wandb

cd ~/cat; git clone https://github.com/lancercat/make_envNG.git

cd make_envNG/ ; sh pylcs.sh ;

unzip pytorch_scatter-laser.zip ; cd pytorch_scatter-2.1.2/; python setup.py install

4. unzip data and model

sh unzip.sh ${DOWNLOADPATH}

5. clone the code

cd ~/cat; git clone https://github.com/lancercat/wna.git

6. stop and check:

```
(catvenv) [lasercat@TESTMEOW ~]$ ls cat
make_envNG  wna
(catvenv) [lasercat@TESTMEOW ~]$ ls hydra_saves
aroute_nd_only-v6S-reg-nedmix-AAF-ohem01E_promelas_b32       pgroute_nd_only-v6S-tiny-AAF
aroute_nd_only-v6S-reg-nedmix-AAF-ohem01E_promelas_mjst_b32  pgroute_nd_only-v6S-tiny-AAF-01E
aroute_nd_only-v6S-tiny-AAF                                  pgroute_nd_only-v6S-tiny-AAF-01E-run2
aroute_nd_only-v6S-tiny-AAF-01E                              pgroute_nd_only-v6S-tiny-AAF-nedmix
aroute_nd_only-v6S-tiny-AAF-01E-run2                         pgroute_nd_only-v6S-tiny-AAF-nedmix-01E
aroute_nd_only-v6S-tiny-AAF-run2                             pgroute_nd_only-v6S-tiny-AAF-nedmix-01E-run2
aroute_nd_only-v6S-tiny-nedmix-AAF                           pgroute_nd_only-v6S-tiny-AAF-nedmix-run2
aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01                    pgroute_nd_only-v6S-tiny-AAF-run2
aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E                   rule_based-v6S-AAF
aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E-run2              rule_based-v6S-AAF-ohem01E
aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01-run2               rule_based-v6S-AAF-ohem01E-run2
aroute_nd_only-v6S-tiny-nedmix-AAF-run2                      rule_based-v6S-AAF-run2
(catvenv) [lasercat@TESTMEOW ~]$ ls ssddata/
anchors        ctwch          CVPR2016  IC13_1015       mlttrchlat_seen     mlttrkr_hori      rctwtrdb_seen_NG
artdb_seen     ctwdb_seen     dicts     IIIT5k_3000     mlttrchlat_seen_NG  NIPS2014          SVT
artdb_seen_NG  ctwdb_seen_NG  dictsv2   lsvtdb_seen     mlttrjp_hori        pami_ch_fsl_hwdb
athenaNG       CUTE80         IC03_867  lsvtdb_seen_NG  mlttrjp_hv          rctwtrdb_seen
```

Reproducing ablative studies

1. run ablative.py (in a screen session as it takes some time):
python ablative.py 2>&1 | tee all.log
(our log is uploaded to github)

---

2. After the abative experiments are finished (usually takes like a dozen hours on an GTX 1650), run ablative2table.py to compute mean and standard deviation:

```
/home/lasercat/catvenv/bin/python3.13 /home/lasercat/cat/wna/ablative2table.py
\newcommand{\LPAoBASEoJPNHVoGZSLoLAoAVG}{40.97}
\newcommand{\LPAoBASEoJPNHVoGZSLoLAoSTD}{1.02}
\newcommand{\LPAoWOAROUTEoJPNHVoGZSLoLAoAVG}{42.11}
\newcommand{\LPAoWOAROUTEoJPNHVoGZSLoLAoSTD}{0.51}
\newcommand{\LPAoPGoLoJPNHVoGZSLoLAoAVG}{41.49}
\newcommand{\LPAoPGoLoJPNHVoGZSLoLAoSTD}{1.01}
\newcommand{\LPAoPGoLoSoJPNHVoGZSLoLAoAVG}{43.09}
\newcommand{\LPAoPGoLoSoJPNHVoGZSLoLAoSTD}{0.33}
\newcommand{\LPAoPGoNoJPNHVoGZSLoLAoAVG}{39.66}
\newcommand{\LPAoPGoNoJPNHVoGZSLoLAoSTD}{0.36}
\newcommand{\LPAoPGoNoSoJPNHVoGZSLoLAoAVG}{43.17}
\newcommand{\LPAoPGoNoSoJPNHVoGZSLoLAoSTD}{0.30}
\newcommand{\LPAoARoLoJPNHVoGZSLoLAoAVG}{41.88}
\newcommand{\LPAoARoLoJPNHVoGZSLoLAoSTD}{0.59}
\newcommand{\LPAoARoLoSoJPNHVoGZSLoLAoAVG}{43.90}
\newcommand{\LPAoARoLoSoJPNHVoGZSLoLAoSTD}{0.86}
\newcommand{\LPAoWOOHEMoJPNHVoGZSLoLAoAVG}{42.05}
\newcommand{\LPAoWOOHEMoJPNHVoGZSLoLAoSTD}{0.25}
\newcommand{\LPAoJPNHVoGZSLoLAoAVG}{45.14}
\newcommand{\LPAoJPNHVoGZSLoLAoSTD}{0.35}
\newcommand{\LPAoOHEMoAoJPNHVoGZSLoLAoAVG}{44.53}
\newcommand{\LPAoOHEMoAoJPNHVoGZSLoLAoSTD}{0.31}
------

Process finished with exit code 0
```

---

3. save the output text as a txt file, say reprod.txt
(our log is can be found in this github repo as reprod.txt)

---

4. diff them:
diff reprod.txt paperref.txt --color

Expect the results to be a bit different here and there due to float error between different implementations of operators, which can be caused by diffferent torch version, gpu structure, vram, etc (yes, torch[1] and CUDNN[2] may choose different backend on different hardware)

[1] https://discuss.pytorch.org/t/different-machines-different-results/100126
[2]https://docs.nvidia.com/deeplearning/cudnn/backend/latest/developer/core-concepts.html

```
(catvenv) [lasercat@TESTMEOW wna]$ diff reprod.txt paperref.txt --color
2c2
< \newcommand{\LPAoBASEoJPNHVoGZSLoLAoSTD}{1.02}
---
> \newcommand{\LPAoBASEoJPNHVoGZSLoLAoSTD}{1.03}
7c7
< \newcommand{\LPAoPGoLoSoJPNHVoGZSLoLAoAVG}{43.09}
---
> \newcommand{\LPAoPGoLoSoJPNHVoGZSLoLAoAVG}{43.07}
9,10c9,10
< \newcommand{\LPAoPGoNoJPNHVoGZSLoLAoAVG}{39.66}
< \newcommand{\LPAoPGoNoJPNHVoGZSLoLAoSTD}{0.36}
---
> \newcommand{\LPAoPGoNoJPNHVoGZSLoLAoAVG}{39.65}
> \newcommand{\LPAoPGoNoJPNHVoGZSLoLAoSTD}{0.35}
13c13
< \newcommand{\LPAoARoLoJPNHVoGZSLoLAoAVG}{41.88}
---
> \newcommand{\LPAoARoLoJPNHVoGZSLoLAoAVG}{41.89}
18,20c18,20
< \newcommand{\LPAoWOOHEMoJPNHVoGZSLoLAoSTD}{0.25}
< \newcommand{\LPAoJPNHVoGZSLoLAoAVG}{45.14}
< \newcommand{\LPAoJPNHVoGZSLoLAoSTD}{0.35}
---
> \newcommand{\LPAoWOOHEMoJPNHVoGZSLoLAoSTD}{0.24}
> \newcommand{\LPAoJPNHVoGZSLoLAoAVG}{45.15}
> \newcommand{\LPAoJPNHVoGZSLoLAoSTD}{0.36}
22a23,24
```

Reproducing Regular Model on OpenSet

run object310-rel/project310_v6SF_stability-re/aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E

```
INFO: starting
/home/lasercat/cat/wna/neko_sdk/neko_framework_NG/modules/concat_mvn_dev.py:67: UserWarning: To copy construct from a tensor, it is recommended to use sourceTensor.detach().clone() or sourceTensor.detach().clone().requires_grad_(True)
  imgt=torch.stack([torch.tensor(i,dtype=this.get_type()) for i in imagelist]).permute(0,3,1,2).contiguous().to(this.mean.data.device)-this.mean;
WARN: Corrupted image for 5171
Date: 2025-06-12 14:36:05.161692 ,TEST: KR-KR-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 5170 ,ACR: 0.195357833655706 ,Lenpred_ACR: 0.6969052224371374 ,FPS: 107.1092715124331
WARN: Corrupted image for 5075
Date: 2025-06-12 14:36:52.248106 ,TEST: JPNHV-JPNHV-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 5074 ,ACR: 0.45309420575482856 ,Lenpred_ACR: 0.8281434765471029 ,FPS: 107.76280253973862
WARN: Corrupted image for 5075
{'Date': datetime.datetime(2025, 6, 12, 14, 37, 38, 995094), 'TEST': 'JPNHV-JPNHV-OSR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.7602158828064765, 'R': 0.7063807254434736, 'P': 0.9487908961593172, 'F': 0.8098345727728031}
WARN: Corrupted image for 5075
{'Date': datetime.datetime(2025, 6, 12, 14, 38, 26, 30309), 'TEST': 'JPNHV-JPNHV-GOSR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.687875574407918, 'R': 0.6534521158129176, 'P': 0.8634490876986463, 'F': 0.7439148073022313}
WARN: Corrupted image for 5075
{'Date': datetime.datetime(2025, 6, 12, 14, 39, 13, 820), 'TEST': 'JPNHV-JPNHV-OSTR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.7174177831912302, 'R': 0.802757564151666, 'P': 0.9109083007388092, 'F': 0.8534201954397395}
Date: 2025-06-12 14:39:46.950327 ,TEST: JPN-JPN-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 4009 ,ACR: 0.4784235470192068 ,Lenpred_ACR: 0.8233973559491145 ,FPS: 118.08836023802422
{'Date': datetime.datetime(2025, 6, 12, 14, 40, 20, 974879), 'TEST': 'JPN-JPN-OSR', 'Epoch': 0, 'Iter': 0, 'Total': 4009.0, 'KACR': 0.7983453981385729, 'R': 0.715318869165023, 'P': 0.958590308370044, 'F': 0.8192771084337348}
{'Date': datetime.datetime(2025, 6, 12, 14, 40, 55, 91256), 'TEST': 'JPN-JPN-GOSR', 'Epoch': 0, 'Iter': 0, 'Total': 4009.0, 'KACR': 0.7238437359676695, 'R': 0.6604938271604939, 'P': 0.8923426838514026, 'F': 0.7591099645275717}
{'Date': datetime.datetime(2025, 6, 12, 14, 41, 28, 923666), 'TEST': 'JPN-JPN-OSTR', 'Epoch': 0, 'Iter': 0, 'Total': 4009.0, 'KACR': 0.7531380753138075, 'R': 0.8235574630424416, 'P': 0.9421713038734315, 'F': 0.878880407124682}
wandb:
wandb: You can sync this run to the cloud by running:
wandb: wandb sync /home/lasercat/hydra_logs/aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E/watch_and_control/aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E/wandb/offline-run-20250612_143501-93lhcncm
wandb: Find logs at: ../../../../../hydra_logs/aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E/watch_and_control/aroute_nd_only-v6S-tiny-nedmix-AAF-ohem01E/wandb/offline-run-20250612_143501-93lhcncm/logs

Process finished with exit code 0
```

Let’s map them to the table

Split	Registered	Out-of-Set	Name	LA	Recall	Precision	FM
GZSL (Hori.)	*Unique Kanji* Shared Kanji, *Kana*, Latin,	∅	OSOCR-Large [4]	30.83	–	–	–
			OpenCCD-Large [23]	41.31	–	–	–
			OpenSAVR-XL [34]	42.58	–	–	–
			MOoSE-XL* [7]	39.56	–	–	–
			CFOR-XL [37]	44.47	–	–	–
			WnA-S (Ours)*	47.89	–	–	–

Date: 2025-06-12 14:39:46.950327 ,TEST: JPN-JPN-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 4009 ,ACR: 0.4784235470192068 ,Lenpred_ACR: 0.8233973559491145 ,FPS: 118.08836023802422

OSTR (Hori.)	Shared Kanji, *Unique Kanji*	*Kana* Latin	OSOCR-Large [4]	58.57	24.46	93.78	38.80
			OpenSAVR-XL [34]	72.33	72.96	92.62	81.62
			MOoSE-XL* [7]	64.80	80.49	89.12	84.50
			CFOR [37]	71.80	**86.36**	89.52	**87.91**
			WnA-S (Ours)*	75.31	82.30	94.21	87.85

6), 'TEST': 'JPN-JPN-OSTR', 'Epoch': 0, 'Iter': 0, 'Total': 4009.0, 'KACR': 0.7531380753138075, 'R': 0.8235574630424416, 'P': 0.9421713038734315, 'F': 0.878880407124682}

GZSL (MO)	Shared Kanji *Unique Kanji*	∅	MOoSE-XL	37.39	–	–	–
			WnA-S (Ours)	45.35	–	–	–

Date: 2025-06-12 14:36:52.248106 ,TEST: JPNHV-JPNHV-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 5074 ,ACR: 0.45309420575482856 ,Lenpred_ACR: 0.8281434765471029 ,FPS: 107.76280253973862

OSR (MO)	Shared Kanji Latin	*Unique Kanji* *Kana*	MOoSE-XL	75.56	**74.05**	**95.79**	**83.53**
			WnA-S (Ours)	76.02	70.63	94.87	80.98

'TEST': 'JPNHV-JPNHV-OSR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.7602158828064765, 'R': 0.7063807254434736, 'P': 0.9487908961593172, 'F': 0.8098345727728031}

GOSR (MO)	Shared Kanji *Unique Kanji*	*Kana*	MOoSE-XL	59.81	63.83	81.89	71.74
			WnA-S (Ours)	68.82	**65.39**	86.35	**74.42**

TEST': 'JPNHV-JPNHV-GOSR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.687875574407918, 'R': 0.6534521158129176, 'P': 0.8634490876986463, 'F': 0.7439148073022313}

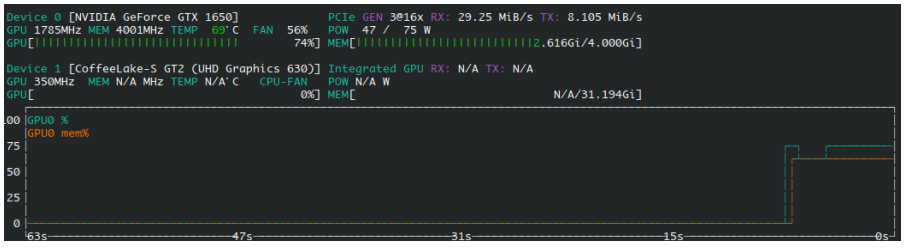
OSTR (MO)	Shared Kanji *Unique Kanji*	Latin *Kana*	MOoSE-XL	61.75	80.01	88.18	83.90
			WnA-S (Ours)	71.78	80.28	91.09	85.34

'TEST': 'JPNHV-JPNHV-OSTR', 'Epoch': 0, 'Iter': 0, 'Total': 5074.0, 'KACR': 0.7174177831912302, 'R': 0.802757564151666, 'P': 0.9109083007388092, 'F': 0.8534201954397395}

GZSL (KR)	*Hangul* Latin	∅	CFOR-XL [37]	22.14	–	–	–
			WnA-S (Ours)	19.52	–	–	–

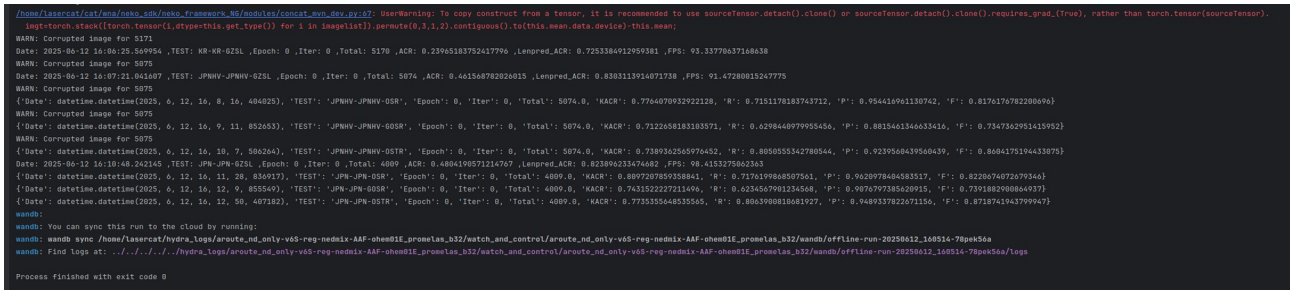
Date: 2025-06-12 14:36:05.161692 ,TEST: KR-KR-GZSL ,Epoch: 0 ,Iter: 0 ,Total: 5170 ,ACR: 0.195357833655706 ,Lenpred_ACR: 0.6969052224371374 ,FPS: 107.1092715124331

Note the performance differs a bit on this 1650, but not much (utilization btw)



Reproducing Large Model on OpenSet

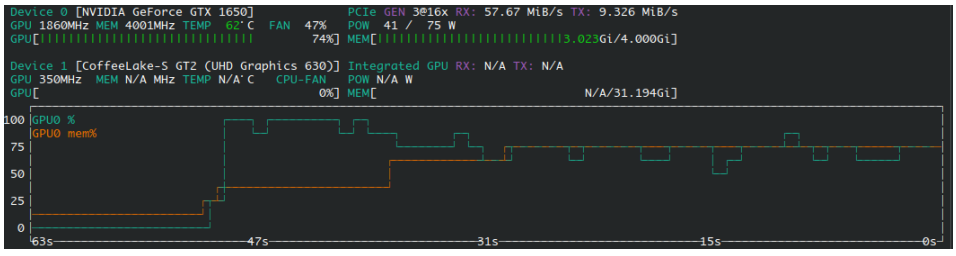
Run object310-rel/XL/aroute_nd_only-v6S-reg-nedmix-AAF-ohem01E_promelas_b32/osocr_benchall.py



Let's map them to the table

Split	Registered	Out-of-Set	Name	LA	Recall	Precision	FM
GZSL (Hori.)	*Unique Kanji* Shared Kanji, *Kana*, Latin,	∅	OSOCR-Large [4]	30.83	–	–	–
			OpenCCD-Large [23]	41.31	–	–	–
			OpenSAVR-XL [34]	42.58	–	–	–
			MOoSE-XL* [7]	39.56	–	–	–
			CFOR-XL [37]	44.47	–	–	–
			WnA-S-XL (Ours)*	**48.02**	–	–	–
OSTR (Hori.)	Shared Kanji, *Unique Kanji*	*Kana* Latin	OSOCR-Large [4]	58.57	24.46	93.78	38.80
			OpenSAVR-XL [34]	72.33	72.96	92.62	81.62
			MOoSE-XL* [7]	64.80	80.49	89.12	84.50
			CFOR [37]	71.80	**86.36**	89.52	**87.91**
			WnA-S-XL(Ours)*	**77.35**	80.68	**94.89**	87.21
GZSL	Shared Kanji		MOoSE-XL	37.39	–	–	–
	Latin, *Kana*		WnA-S-XL(Ours)	**46.14**	–	–	–
OSR	Shared Kanji	*Unique Kanji*	MOoSE-XL	75.56	**74.05**	**95.79**	**83.53**
			WnA-S-XL(Ours)	**77.64**	71.45	95.44	81.72
GOSR	Shared Kanji	*Kana*	MOoSE-XL	59.81	63.83	81.89	71.74
	Latin		WnA-S-XL(Ours)	**71.22**	62.89	**88.08**	73.38
OSTR	Shared Kanji	Latin	MOoSE-XL	61.75	80.01	88.18	83.90
			WnA-S-XL(Ours)	**73.89**	**80.50**	**92.39**	**86.04**
GZSL	*Hangul*	∅	CFOR-XL [37]	22.14	–	–	–
			WnA-S-XL (Ours)	**24.00**	–	–	–

Note the performance differs a bit on this 1650, but not much (utilization btw)



Reproducing Fig 1& Inference on your own data

Training from Scratch